



IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

In re application of:
Bentley et al.

Appl. No. 09/929,278

Filed: August 15, 2001

For: COMPUTER READABLE MEDIUM
FOR STORING LARGE DATA FILES

Art Unit: 2167

Examiner: Miranda Le

Atty. Docket No. 36488-169754

Customer No.

26694

PATENT TRADEMARK OFFICE

Response to Notice of Non-Compliant Appeal Brief

MS Appeal Brief
Assistant Commissioner for Patents
P.O. Box 1450
Alexandria, VA 22313-1450

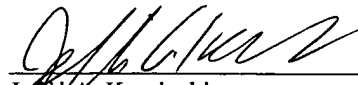
Sir:

In response to Notice of Non-Compliant Appeal Brief dated February 5, 2008, Applicants resubmit a copy of the Appeal Brief of July 5, 2007. Submitted herewith is a corrected section of the Appeal Brief to the Claim Appendix as stated in the notice dated February 5, 2008.

As respectfully submitted that the Appeal Brief is compliance with 37 C.F.R. 41.37(c). Entry of the Appeal Brief is respectfully requested. Please do not hesitate to contact the undersigned by telephone if there are any additional problems with the amendment.

Respectfully submitted,

Date: 2/26/08


Jeffrey A. Kaminski
Registration No. 42,709
VENABLE
P.O. Box 34385
Washington, D.C. 20043-9998
Telephone: (202) 344-4000
Telefax: (202) 344-8300

/lrh



Docket No.: 36488-169754
(PATENT)

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

In re Patent Application of:
Bentley et al.

Art Unit: 2167

Application No: 09/929,278

Examiner: Miranda Le

Confirmation No: 8202

Filed: August 15, 2001

Atty. Docket No: 36488-169754

For: COMPUTER READABLE MEDIUM FOR
STORING LARGE DATA FILES

Customer No:

26694

PATENT TRADEMARK OFFICE

APPEAL BRIEF

MS Appeal Brief - Patents
Commissioner for Patents
P.O. Box 1450
Alexandria, VA 22313-1450

Dear Sir:

In response to the Notice of Non-Compliant Appeal Brief of October 17, 2007,
Applicants resubmit the following Appeal Brief. Should any additional fees be due, please charge
our deposit account no. 22-0261.

A copy of the Notice is attached.

As required under § 41.37(a), this brief is filed within two months of the Notice of
Appeal filed in this case on May 8, 2007, and is in furtherance of said Notice of Appeal.

The fees required under § 41.20(b)(2), and any required petition for extension of time for filing this brief and fees therefore, are dealt with in the accompanying TRANSMITTAL OF APPEAL BRIEF.

This brief contains items under the following headings as required by 37 C.F.R. § 41.37 and M.P.E.P. § 1206:

I.	Real Party In Interest
II	Related Appeals and Interferences
III.	Status of Claims
IV.	Status of Amendments
V.	Summary of Claimed Subject Matter
VI.	Grounds of Rejection to be Reviewed on Appeal
VII.	Argument
Appendix A	Claims
Appendix B	Evidence
Appendix C	Related Proceedings

I. REAL PARTY IN INTEREST

The real party in interest for this appeal is:

Bentley System, Inc.
685 Stockton Drive
Exton, PA 19341-0678

II. RELATED APPEALS, INTERFERENCES, AND JUDICIAL PROCEEDINGS

There are no other appeals, interferences, or judicial proceedings which will directly affect or be directly affected by or have a bearing on the Board's decision in this appeal.

III. STATUS OF CLAIMS

A. Total Number of Claims in Application

There are 34 claims pending in application.

Claims 83-116 have been finally rejected in the Office Action of February 8, 2007.
Claims 83-116 are being appealed. Claims 1-82 have been canceled.

IV. STATUS OF AMENDMENTS

No claim amendments were made after final rejection.

V. SUMMARY OF CLAIMED SUBJECT MATTER

As recited in claim 83, a file format 200 uses variable-sized data records, or elements, stored in element chunks 320, to store data in a file on a computer readable medium (specification page 6, lines 29-31, page 7 lines 1-2, FIG. 8b steps 456-466). When an element is modified, the element may be stored in its element chunk 320, which may be written to the computer readable medium without requiring the rewriting of the entire file in which the element chunk 320 is stored, and without moving the element chunk 320 to the end of the file leaving a hole in the file (specification, page 11, lines 31, FIG. 8b, steps 458-466). Instead, only the element chunk 320 containing the modified element is rewritten in the computer readable storage medium (specification page 11, lines 11-14). Additionally, the element chunk 320 may be compressed and/or encrypted before it is written to the computer readable medium, without requiring the recompression or re-encryption of the entire file in which the element chunk 320 is stored (specification page 11, lines 11-14).

Claim 83 recites: a computer readable medium for storing data for access by an application program, comprising:

a file format 200 defining a structure of a file stored in said computer readable medium, the file format including (specification, page 6, lines 29-31; page 7, lines 1-2; page 8, lines 22-26; FIG. 3)

elements stored in the computer readable medium, the elements being variable sized data records arranged in a format that can be interpreted by a computer program (specification, page 7, lines 1-2; page 9, lines 22-26);

element chunks 320 stored in the computer readable medium, the element chunks 320 being variable sized and including groups of the elements, the element chunks 320 having a unique name and a fixed header 325 including at least one of a number of elements in the element chunk 320, a compression scheme, or an encryption scheme for the elements (specification, page 11, lines 3-14, 27-31; page 12, lines 1-5, 13-31; page 13, lines 10-22; FIG. 5, 320; FIG. 7, 320);

a model 310 stored in the computer readable medium, the model 310 including groups of related element chunks 320 and a model 314 header stream, the model header stream 314 including at least one of a model name, units, or a geometric range for the model (specification, page 9, lines 23-30; page 10, lines 17-22; FIG. 4, 310, 314, 320; FIG. 5, 310); and

a root storage 210 stored in the computer readable medium, the root storage 210 including at least one model 310 (specification, page 9, lines 3-5, 26-30; page 12, lines 3-4; FIG. 3, 210, 300).

Claim 99 recites: a computer program product comprising a computer readable medium having a computer program logic stored therein, the computer program logic comprising:

means for enabling said computer system to allocate elements having a variable size to element chunks 320, the element chunks 320 being variable sized and including groups of the elements, the element chunks 320 having a unique name and a fixed header 325 including at least one of a number of elements in the element chunk 320, a compression scheme, or an encryption scheme for the elements (specification, page 11, lines 3-14, 27-31; page 12, lines 1-5, 13-31; page 13, lines 10-22; FIG. 5, 320; FIG. 7, 320);

means for enabling said computing unit to store in the computer readable medium at least one model 310, wherein said at least one model 310 is for grouping related elements, is identifiable by a unique identifier, and comprises a control element list having variable sized element chunks 320 containing control elements, and a graphic element list having variable sized element chunks 320 containing graphic elements (specification, page 9, lines 23-30; page 10, lines 17-22; FIG. 4, 310, 314, 320; FIG. 5, 310); and

means for enabling a computing unit to store a root storage 210 comprising the model 310 in the computer readable medium (specification, page 9, lines 3-5, 26-30; page 12, lines 3-4; FIG. 3, 210, 300).

Claim 101 recites: a CAD design file having a file format 200 and stored on a computer readable medium, the CAD design file comprising:

elements representing items of the CAD design, the elements being variable sized data records arranged in a format that can be interpreted by a computer program (specification, page 7, lines 1-2; page 9, lines 22-26; page 10, lines 25-29),

element chunks 320 including groups of the elements, the element chunks 320 having a unique name and a fixed header 325 including at least one of a number of elements in the element chunk 320, a compression scheme, or an encryption scheme for the elements, the element chunks 320 having a variable size, the groups of elements including control elements having no physical representation and graphic elements having a graphical representation (specification, page 11, lines 3-14, 27-31; page 12, lines 1-5, 13-31; page 13, lines 10-22; FIG. 5, 320; FIG. 7, 320);

a model 310, the model 310 including groups of related element chunks 320 and a model header stream 314, the model header stream 314 including at least one of a model 310 name, units, or a geometric range for the model 310 (specification, page 9, lines 23-30; page 10, lines 17-22; FIG. 4, 310, 314, 320; FIG. 5, 310); and

a root storage 210 including at least one model 310 and a control model 310 storing information shared across other models 310 in the root storage 210 (specification, page 9, lines 3-5, 26-30; page 12, lines 3-4; FIG. 3, 210, 300).

Embodiments of the invention relate to a computer readable medium and CAD design file having a format for storing data for access by an application program. These embodiments provide numerous advantages over known computer readable mediums for storing data for access by an application program. Such advantages are described in the specification at page 6, lines 27-29, and page 11, lines 11-26, for example. The advantages include allowing large quantities of variable-sized data records to be stored on a storage medium in an enhanced file format 200 and to allowing efficient access and control over data stored in the enhanced file format 200. For example, one of the variable-sized data records stored on a storage medium in a file of the enhanced file format 200

may be modified without requiring the entire file to be rewritten and without creating wasted space within the file.

These and other advantages are achievable with the embodiments recited, for example, in independent claim 83. As recited in independent claim 83, a computer readable medium for storing data for access by an application program is provided. In conventional computer readable mediums, data records stored in files must have a maximum size. If a data record is modified to exceed to this maximum size, either it must be rewritten at the end of the file, leaving a hole in the file where the data record was previously written, or the entire file must be rewritten to accommodate the increased size of the data record in the position where the data record was previously written. The modified data record may be compressed, but this still requires rewriting the entire file, as the entire file will need to be uncompressed and recompressed in order for the modified data record to be compressed.

Accordingly, the embodiments allow for the variable-sized data records in a file stored on a computer readable medium to be modified, compressed and/or encrypted without requiring the entire file be rewritten, recompressed, or re-encrypted, and without leaving holes in the file due to data records being moved to the end of the file.

VI. GROUNDS OF REJECTION TO BE REVIEWED ON APPEAL

Whether the Examiner has established that claims 83-94 and 99-112 are obvious in light of U.S. Patent Number 6,718,329 to Selvin in view of U.S. Pub. No. 20010004245 from Shiba, and claims 95-98 and 113-116 are obvious in light of Selvin in view of Shiba in further view of U.S. Pub. No. 20020161608 from Loveland.

VII. ARGUMENT

Claims 83-116 recite a file format for storing a specific type of data (variable length CAD elements), that is carefully crafted and designed to facilitate a specific usage pattern (optimized saving of incremental changes to a subset of CAD elements in a large file). As is known

to one of ordinary skill in the art, a file when used in connection with computer systems is an entity of data available to users of the system that can be manipulated as an entity. For example, a file can be moved from computer to computer, directory to directory, transmitted over a network, etc. A file may be, for example, a Word document, which may be attached to an email and transmitted over a network to a recipient. All of the claims of the present application describe what is in the file and the format of the file. The cited art does not teach nor relate to a file format at all, and is related to the claimed invention only at a superficial level by virtue of common generic computer terminology as detailed below.

Selvin describes a method for generating nodes and links for a hypertext database from a source document (Selvin, abstract). A source document is any data file that contains its data in text format (Selvin, col. 4, lines 21-26). Such a source document is divided into segments based on its format properties (Selvin, col. 4, lines 35-57, col. 5, lines 64-67, col. 6, lines 1-4). Nodes are generated to represent each segment, and links definitions are generated for links that will connect the generated nodes, so that the meaning of a link between two nodes can be determined from the link's definition (Selvin, col. 6, lines 4-6). The generated nodes and links are imported into a hypertext database, where they can be viewed with an application program capable of viewing hypertext (Selvin, col. 6, lines 6-9). For example, FIG. 4 of Selvin demonstrates a source document that has undergone the process of having nodes and links generated from it. The links and nodes that are the result of the process form a structural representation of the original source document that is suitable for conversion into hypertext data (Selvin, FIG. 4, 60, 60A, 60B, 60C, 70).

Selvin does not disclose or suggest all of the elements of the present claims. Selvin is not in the same area as the present application, as Selvin does not disclose or suggest a new file format of any kind, as in claim 83, a computer readable medium comprising instructions for storing data on a computer readable medium, as in claim 99, or CAD file having a new file format, as in claim 101. Selvin relies on existing file formats, for example, HTML documents, Microsoft Word and Excel documents, and CAD documents, but does not suggest any new manner in which the data in these documents may be stored on a computer readable medium (Selvin, col. 4, lines 19-34). The nodes and links generated from the source document do not constitute a file format, as suggested in

the Action on page 2, paragraph 4. Instead, the nodes and links are data that may reference data in another previously existing file. It may be possible to store nodes and links in files but certainly they do not in any way define a file format. In fact, Selvin's mention of writing nodes to a file (Selvin col 9, lin3 45) is merely a transient text-file mechanism to import the data to a database.

In contrast to Selvin, claim 83 of the present application discloses a new file format for storing data on a computer readable medium, one that is not a hypertext database. Selvin only discloses specific instances of using extant file formats, showing only that many file formats exist. Claim 83 recites the structure of a new file format that is not of the previous types, including hypertext databases. A file format, such as the file format recited in claim 83, governs the way in which the physical properties of a computer readable storage medium are manipulated in order to store data, whereas in Selvin all that is disclosed is a type of data structure. A data structure, such as that in Selvin, discloses nothing about how the data structure is physically stored on a computer readable storage medium. Selvin merely describes data that is stored in a hypertext database.

The Action, on page 2, section 4, further aligns the hypertext database of Selvin with "defining a structure of a file stored in said computer readable medium." Again, Selvin has provided only a data type which may be stored on said computer readable medium, in this case, a hypertext database. Nowhere does Selvin disclose an actual structure for the hypertext database as it is stored on a computer readable medium (Selvin, col. 9, 16-20, FIG. 1, 44). This is wholly different from defining a structure of a file stored in said computer readable medium. Claim 83, in contrast to Selvin, recites a new structure for a file stored on a computer readable medium. This structure may define the way in which any data type may be written on to a computer readable medium, governing the way in which the physical properties of the computer readable medium are manipulated to represent data.

The file format recited in claim 83 includes elements stored in the computer readable medium, the elements being variable sized data records. The Action, on page 3, aligns nodes and inter-node associations with elements stored in the computer readable medium, and suggests that the use of the word "length" discloses elements being variable sized data records (Selvin, col. 11, lines 46-59, col. 7, lines 22-43). Elements, as defined in claim 83, are variable sized data records stored on a computer readable medium (specification, page 7, lines 1-2). Nodes and inter-node

associations are data, but they are not elements themselves as they do not define a file format structure on a computer readable medium. Selvin makes no mention of the format in which nodes and inter-node associations are stored on a computer readable medium. Elements may store within them, on a computer readable medium, a wide variety of data types. Nodes and inter-node associations are data types and do not store other data types.

Further, “length,” as disclosed in Selvin, only pertains to the “LABEL field” of a node (Selvin, col. 7, 24-25). The length of said “LABEL field” of a node is “delimited by length or the format codes of source document 42” (Selvin, col. 7, lines 24-25). This is wholly different from having variable sized data records for elements. Selvin only discloses that some feature of the source document may delimit the length of one of the pieces of information contained within a node, the LABEL. In contrast, the variable sized data records recited in claim 83 are data records used to store elements and the length field of an Element is necessary to determine its size in bytes. It would make no sense to map the optional LABEL field mentioned in Selvin to the length of an Element, since one there cannot be an Element in the present invention without its length.

Claim 83 further recites element chunks stored in the computer readable medium, the element chunks being variable sized and including groups of the elements, the element chunks having a unique name, a fixed header including at least one of an number of elements in the element chunks. The Action on page 3 aligns element chunks with the LABEL field, CONTENT field and NODE TYPE field of Selvin. This contradicts the Action’s previous alignment of elements with nodes and inter-node associations, as discussed above. LABEL field, CONTENT field, and NODE TYPE field are all data fields stored within a node (Selvin, col.7, lines 21-42). Element chunks, as recited in claim 83, contain elements within them. Hence, if nodes and inter-node associations are aligned with elements, then in order for LABEL field, CONTENT field, and NODE TYPE field to be aligned with element chunks, the LABEL field, CONTENT field, and NODE TYPE field must contain nodes and inter-node associations, as element chunks contain elements. This is the opposite of what is disclosed in Selvin. Further, LABEL field, CONTENT field, and NODE TYPE field do not, independently of their relation to nodes, align with element chunks. Element chunks, as recited in claim 83, contain elements, which may store data of any type on the computer readable medium. An element chunk may contain as many elements as will fit into the size of the element chunk, and

each element in the same element chunk may represent any type of data that an element may represent.

In contrast, the LABEL field, CONTENT field, and NODE TYPE field may each contain only a specific type of data. For example, a CONTENT field “contains the format codes and the remaining content of the paragraph” (Selvin, col. 7, lines 32-33). These restrictions on data type are not present in element chunks. Further still, the fixed header of an element chunk contains data relating to the elements stored within the element chunk, for example, how many elements there are and how they are encrypted or compressed, allowing for the reading of those elements off of the computer readable medium (specification, page 12, lines 13-21). The “heading” disclosed in Selvin is not actually a heading in and of itself. It is presented in Selvin as an example of the type of indicator that might be stored in the LABEL field of a node to help define the nodes relation to other nodes. The data in the LABEL field, i.e., “heading,” plays no role in assisting with the reading of the contents of the node, and cannot contain data about the quantity of data stored in the node, or whether or not it is encrypted or compressed. The LABEL field may contain “heading”, “paragraph”, “heading 1”, “paragraph 1”, and so forth, as it is merely a descriptor of the node. (Selvin, col. 7, lines 22-43), and therefore is wholly different from the fixed header recited in claim 83.

Claim 83 also recites a model stored in the computer readable medium, the model including groups of related element chunks and a model header stream, the model header stream including at least one of a model name, units, or a geometric range for the model. Selvin does not disclose or suggest any of these features. The Action on page 3 aligns the “link-type” from Selvin with the model recited in claim 83. Link-types in Selvin are applied to links already generated between nodes, and serve only to “define further desired relationships between linked nodes” (Selvin, col. 11, lines 58-59). This has no relationship to a model as recited in claim 83, which may be a collection of elements that have been grouped together to be stored on a computer readable medium (specification, page 9, lines 23-30). As recited in claim 83, a model may be a structure written to a computer readable medium that contains element lists, which may be structures on the a computer readable medium which may contain element chunks, which may in turn be structures on the computer readable medium containing elements (FIG. 5, 310, 315, 320). Link-types do not

contain anything aside from information further defining the link to which a link-type pertains (Selvin, col. 11 lines 59-59, col. 12, lines 2-8). Link-types do not define a structure for storing data on a computer readable medium, as they are just data which may be stored on a computer readable medium in any appropriate format. Link-types, as disclosed in Selvin, are a wholly different concept from a model as recited in claim 83.

The Action further aligns “heading” from Selvin with the model header stream recited in claim 23. As discussed above, “heading” in Selvin is merely an example given for the type of information that may be found in the LABEL field of the node, which may help to define the nodes relationship to other nodes (Selvin, col. 7, lines 22-43). As recited in claim 83, a model header stream may contain information specific to a model including at least one of the model name, the units, and the geometric range of the model, as well as any other information relevant to the model (specification, page 10, lines 12-22). A model header stream is therefore capable of storing more and different types of information than the LABEL field of a node in Selvin. Further, Selvin does not disclose or suggest anything similar to a model name, units or a geometric range being contained in the LABEL field. All that may be contained in the LABEL field for a node is the label, i.e., “heading”, “heading1”, etc., for that node that defines that node’s relationship to other nodes (Selvin, col. 7, lines 22-43). These labels are relational descriptors and not units, as no units of any type are suggested, geometric ranges, as nothing similar to a geometric range is suggested, or model names, as the label must describe a relationship between nodes.

Claim 83 further recites a root storage stored in the computer readable medium, the root storage including at least one model. This is also not disclosed or suggested by Selvin. The Action on page 3 aligns “a node” with a root storage. A node, as suggested in Selvin, is wholly different from a root storage. A node is a data structure which contains data and may include fields which specify particular properties or characteristics of the data (Selvin, col. 4 lines 58-63). A root storage is not a data structure, but is part of file format used in storing a data structure on a computer readable medium. A root storage is the starting point for reading or creating files on the computer readable medium in the file format disclosed in claim 83 (specification, page 9, lines 5-10). As such, a root storage may be used in a file on a computer readable medium which may contain a node of the type suggested in Selvin. However, a node as suggested in Selvin may not contain a root

storage, because a node is a data structure that contains data, and a root storage is not, in itself, data. Therefore, a node and a root storage must be wholly different. If they disclosed the same concept or feature, a root storage's ability to be used in a file format storing a node on a computer readable medium would also mean that a node would be capable of being used in a file format for storing a root storage on a computer readable medium. However, a node discloses nothing about the manipulation of the physical properties of a computer readable storage medium to store data.

The Action on page 3 states that Selvin does not disclose or suggest the features recited in claim 83 of a compression scheme, or an encryption scheme for the elements. Shiba is cited as disclosing these features (Shiba, paragraph 27). Thus, Shiba is not cited as and does not cure the above-described deficiencies of Selvin.

With regards to independent claim 99, the Action cites the same sections of the prior art as discussed above for independent claim 83, so the arguments made above apply. The combination of Selvin and Shiba does not disclose or suggest any of the features of claim 99.

With regards to independent claim 101, the Action cites the same sections of the prior art as discussed above for claim 83, so the arguments made above apply. Further, Selvin does not disclose or suggest a CAD design having a file format, as stated in the Action on page 5. Selvin only mentions the existence of CAD programs in passing, nowhere in Selvin is a file format for storing a CAD file on computer readable medium disclosed. Selvin mentions that a "source document" may be able to be edited by a CAD program, implying that a "source document" may be a CAD file, which may then be stored in a hypertext database in a nodes and links data structure (Selvin, col. 4, lines 21-57). However, if anything, this may only imply a new data structure for a CAD file, not a new file format, as data structures and file formats are distinct concepts, as discussed above. The combination of Selvin and Shiba does not disclose or suggest any of the features of claim 99.

Further, please see U.S. Patent No. 7,162,479 for Method and System for Storing Large Data Files, which issued to present Applicant on January 9, 2007. The claims from U.S. Patent No. 7,162,479 cover subject matter similar to the subject matter covered by the claims of the present application, and their allowance is an indicator of the allowability of the claims of the present application.

In view of the above discussion, it is clear that the cited reference fails to teach or suggest the features recited in the claims. Each of the dependent claims is allowable for at least the reasons as being dependent from an allowable independent claim. For these reasons, the withdrawal of the rejections of claims 83-116 is respectfully requested.

Dated: 2/26/08

Respectfully submitted,

By 

Jeffri A. Kaminski

Registration No.: 42,709

James R. Burdett

Registration No.: 31,594

VENABLE LLP

P.O. Box 34385

Washington, DC 20043-9998

(202) 344-4000

(202) 344-8300 (Fax)

Attorney/Agent For Applicant

CLAIMS APPENDIX

1-82 (Canceled)

83. (Previously Presented) A computer readable medium for storing data for access by an application program, comprising:

a file format defining a structure of a file stored in said computer readable medium, the file format including,

elements stored in the computer readable medium, the elements being variable sized data records arranged in a format that can be interpreted by a computer program,

element chunks stored in the computer readable medium, the element chunks being variable sized and including groups of the elements, the element chunks having a unique name and a fixed header including at least one of a number of elements in the element chunk, a compression scheme, or an encryption scheme for the elements,

a model stored in the computer readable medium, the model including groups of related element chunks and a model header stream, the model header stream including at least one of a model name, units, or a geometric range for the model, and

a root storage stored in the computer readable medium, the root storage including at least one model.

84. (Previously Presented) The computer readable medium of claim 83, further comprising element lists including element chunks classified according to their meaning in the model, the element lists including the unique name for each element chunk in the respective element list.

85. (Previously Presented) The computer readable medium of claim 84, wherein the elements include control elements having no physical representation and graphic elements having a graphical representation.

86. (Previously Presented) The computer readable medium of claim 85, wherein the element lists include a graphic element list listing the graphic elements and a control element list listing the control elements.

87. (Previously Presented) The computer readable medium of claim 83, wherein the element chunks include a fixed number of elements.

88. (Previously Presented) The computer readable medium of claim 83, further comprising:

- a plurality of models; and
- a model directory stored in the root storage and including a list of the models, the models having a unique name within their respective model directory.

89. (Previously Presented) The computer readable medium of claim 83, wherein the root storage further includes a file header stream, a session information stream, a manifest information stream, or a file properties stream stored therein.

90. (Previously Presented) The computer readable medium of claim 83, further comprising a control model directly stored in the root storage and storing information shared across other models in the root storage.

91. (Previously Presented) The computer readable medium of claim 83, wherein said root storage further comprises a control model storage containing a control model header, a global control element list storage and a global graphic element list storage, wherein said global control element list storage and said global graphic element list storage contain element chunk including global elements.

92. (Previously Presented) The computer readable medium of claim 91, wherein said global elements contain information relevant for all models in said model directory storage.

93. (Previously Presented) The computer readable medium of claim 86, wherein at least one element chunk in said graphic element list is compressed.
94. (Previously Presented) The computer readable medium of claim 86, wherein at least one element chunk in said control element list is compressed.
95. (Previously Presented) The computer readable medium of claim 86, wherein at least one element chunk in said graphic element list is encrypted.
96. (Previously Presented) The computer readable medium of claim 86, wherein at least one element chunk in said control element list is encrypted.
97. (Previously Presented) The computer readable medium of claim 86, wherein at least one element chunk in said control element list is encrypted and compressed.
98. (Previously Presented) The computer readable medium of claim 86, wherein at least one element chunk in said graphic element list is encrypted and compressed.
99. (Previously Presented) A computer program product comprising a computer readable medium having a computer program logic stored therein, the computer program logic comprising:
means for enabling said computer system to allocate elements having a variable size to element chunks, the element chunks being variable sized and including groups of the elements, the element chunks having a unique name and a fixed header including at least one of a number of elements in the element chunk, a compression scheme, or an encryption scheme for the elements;
means for enabling said computing unit to store in the computer readable medium at least one model, wherein said at least one model is for grouping related elements, is identifiable by a unique identifier, and comprises a control element list having variable sized element chunks containing control elements, and a graphic element list having variable sized element chunks containing graphic elements; and

means for enabling a computing unit to store a root storage comprising the model in the computer readable medium.

100. (Previously Presented) The computer program product of claim 99, further comprising:

means for compressing each element chunk stream to be stored in said graphic element list storage or said control model list storage in said control model directory, means for enabling said computer system to store a graphic element list storage and a control element list storage in each control model.

101. (Previously Presented) A CAD design file having a file format and stored on a computer readable medium, the CAD design file comprising:

elements representing items of the CAD design, the elements being variable sized data records arranged in a format that can be interpreted by a computer program,

element chunks including groups of the elements, the element chunks having a unique name and a fixed header including at least one of a number of elements in the element chunk, a compression scheme, or an encryption scheme for the elements, the element chunks having a variable size, the groups of elements including control elements having no physical representation and graphic elements having a graphical representation,

a model, the model including groups of related element chunks and a model header stream, the model header stream including at least one of a model name, units, or a geometric range for the model, and

a root storage including at least one model and a control model storing information shared across other models in the root storage.

102. (Previously Presented) The computer readable medium of claim 101, further comprising element lists including element chunks classified according to their meaning in the model, the element lists including the unique name for each element chunk in the respective element list.

103. (Previously Presented) The computer readable medium of claim 102, wherein the elements include control elements having no physical representation and graphic elements having a graphical representation.

104. (Previously Presented) The computer readable medium of claim 103, wherein the element lists include a graphic element list listing the graphic elements and a control element list listing the control elements.

105. (Previously Presented) The computer readable medium of claim 101, wherein the element chunks include a fixed number of elements.

106. (Previously Presented) The computer readable medium of claim 101, further comprising:

a plurality of models; and

a model directory stored in the root storage and including a list of the models, the models having a unique name within their respective model directory.

107. (Previously Presented) The computer readable medium of claim 101, wherein the root storage further includes a file header stream, a session information stream, a manifest information stream, or a file properties stream stored therein.

108. (Previously Presented) The computer readable medium of claim 101, further comprising a control model directly stored in the root storage and storing information shared across other models in the root storage.

109. (Previously Presented) The computer readable medium of claim 101, wherein said root storage further comprises a control model storage containing a control model header, a global control element list storage and a global graphic element list storage, wherein said global control

element list storage and said global graphic element list storage contain element chunk including global elements.

110. (Previously Presented) The computer readable medium of claim 109, wherein said global elements contain information relevant for all models in said model directory storage.

111. (Previously Presented) The computer readable medium of claim 104, wherein at least one element chunk in said graphic element list is compressed.

112. (Previously Presented) The computer readable medium of claim 104, wherein at least one element chunk in said control element list is compressed.

113. (Previously Presented) The computer readable medium of claim 104, wherein at least one element chunk in said graphic element list is encrypted.

114. (Previously Presented) The computer readable medium of claim 104, wherein at least one element chunk in said control element list is encrypted.

115. (Previously Presented) The computer readable medium of claim 104, wherein at least one element chunk in said control element list is encrypted and compressed.

116. (Previously Presented) The computer readable medium of claim 104, wherein at least one element chunk in said graphic element list is encrypted and compressed.

EVIDENCE APPENDIX

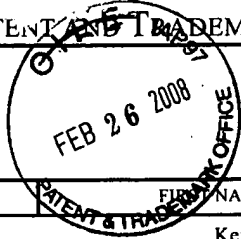
No evidence pursuant to §§ 1.130, 1.131, or 1.132 or entered by or relied upon by the examiner is being submitted.

RELATED PROCEEDINGS APPENDIX

No related proceedings are referenced in II. above, or copies of decisions in related proceedings are not provided.



UNITED STATES PATENT AND TRADEMARK OFFICE



UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRM NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
09/929,278	08/15/2001	Keith Bentley	36488-169754	8202

7590 02/05/2008

Venable
P.O. Box 34385
Washington, DC 20043-9998

EXAMINER

ART UNIT PAPER NUMBER

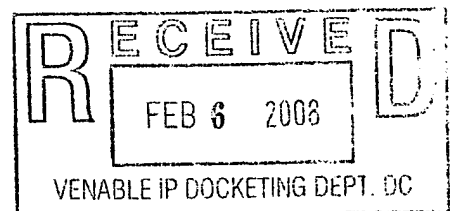
DATE MAILED: 02/05/2008

Please find below and/or attached an Office communication concerning this application or proceeding.

DOCKETED

*Reply to notice of non-compliance
Appeal Brief*

CLIENT/MATTER # _____ ATTY _____
DUE DATE 3/6/08
FINAL DEADLINE 8/5/08
DKTED BY Cut



**Notification of Non-Compliant Appeal Brief
(37 CFR 41.37)**

Application No.

09/929,278

Examiner

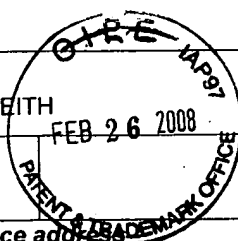
Miranda Le

Applicant(s)

BENTLEY, KEITH

Art Unit

2167



--The MAILING DATE of this communication appears on the cover sheet with the correspondence address--

The Appeal Brief filed on 09 November 2007 is defective for failure to comply with one or more provisions of 37 CFR 41.37.

To avoid dismissal of the appeal, applicant must file an amended brief or other appropriate correction (see MPEP 1205.03) within **ONE MONTH or THIRTY DAYS** from the mailing date of this Notification, whichever is longer.
EXTENSIONS OF THIS TIME PERIOD MAY BE GRANTED UNDER 37 CFR 1.136.

1. ☐ The brief does not contain the items required under 37 CFR 41.37(c), or the items are not under the proper heading or in the proper order.
2. ☒ The brief does not contain a statement of the status of all claims, (e.g., rejected, allowed, withdrawn, objected to, canceled), or does not identify the appealed claims (37 CFR 41.37(c)(1)(iii)).
3. ☐ At least one amendment has been filed subsequent to the final rejection, and the brief does not contain a statement of the status of each such amendment (37 CFR 41.37(c)(1)(iv)).
4. ☐ (a) The brief does not contain a concise explanation of the subject matter defined in each of the independent claims involved in the appeal, referring to the specification by page and line number and to the drawings, if any, by reference characters; and/or (b) the brief fails to: (1) identify, for each independent claim involved in the appeal and for each dependent claim argued separately, every means plus function and step plus function under 35 U.S.C. 112, sixth paragraph, and/or (2) set forth the structure, material, or acts described in the specification as corresponding to each claimed function with reference to the specification by page and line number, and to the drawings, if any, by reference characters (37 CFR 41.37(c)(1)(v)).
5. ☐ The brief does not contain a concise statement of each ground of rejection presented for review (37 CFR 41.37(c)(1)(vi)).
6. ☐ The brief does not present an argument under a separate heading for each ground of rejection on appeal (37 CFR 41.37(c)(1)(vii)).
7. ☐ The brief does not contain a correct copy of the appealed claims as an appendix thereto (37 CFR 41.37(c)(1)(viii)).
8. ☐ The brief does not contain copies of the evidence submitted under 37 CFR 1.130, 1.131, or 1.132 or of any other evidence entered by the examiner **and relied upon by appellant in the appeal**, along with a statement setting forth where in the record that evidence was entered by the examiner, as an appendix thereto (37 CFR 41.37(c)(1)(ix)).
9. ☐ The brief does not contain copies of the decisions rendered by a court or the Board in the proceeding identified in the Related Appeals and Interferences section of the brief as an appendix thereto (37 CFR 41.37(c)(1)(x)).
10. ☐ Other (including any explanation in support of the above items):

Miranda Le
February 01, 2008